

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

MULTIPROCESSOR ARITHMETIC UNIT AND PROGRAMMABLE CONTROLLER HAVING THE ARITHMETIC UNIT

Patent Number: JP8249022

Publication date: 1996-09-27

Inventor(s): NISHIMURA YASUHIRO

Applicant(s):: OMRON CORP

Requested Patent: JP8249022

Application Number: JP19950052286 19950313

Priority Number(s):

IPC Classification: G05B19/05 ; G05B15/02

EC Classification:

Equivalents:

Abstract

PURPOSE: To reduce the arithmetic processing overhead at the slave processor side of a multiprocessor arithmetic unit.

CONSTITUTION: A decoding part 11 analyzes every instruction and outputs an arithmetic command to a master processor 1 or a slave processor 2 together with an instruction operand based on the interpretation of analyses of instructions. An address calculation part 12a calculates an address based on the instruction operand and stores the address in an address register 14a. A slave arithmetic part 21 of the processor 2 receives an arithmetic command from the part 11 and has an access to an interface part 14. Then an interface controller 14b performs the access control based on the address stored in the register 14a, so that the part 21 can have an access to the address data stored in a data memory 4.

Data supplied from the esp@cenet database - I2

(51) Int.Cl.⁶
G 0 5 B 19/05
15/02

識別記号 庁内整理番号
7531-3H

F I
G 0 5 B 19/05
15/02
19/05

技術表示箇所
S
M
G

審査請求 未請求 請求項の数4 OL (全8頁)

(21)出願番号

特願平7-52286

(22)出願日

平成7年(1995)3月13日

(71)出願人 000002945

オムロン株式会社

京都府京都市右京区花園土堂町10番地

(72)発明者 西村 康裕

京都府京都市右京区花園土堂町10番地 オ
ムロン株式会社内

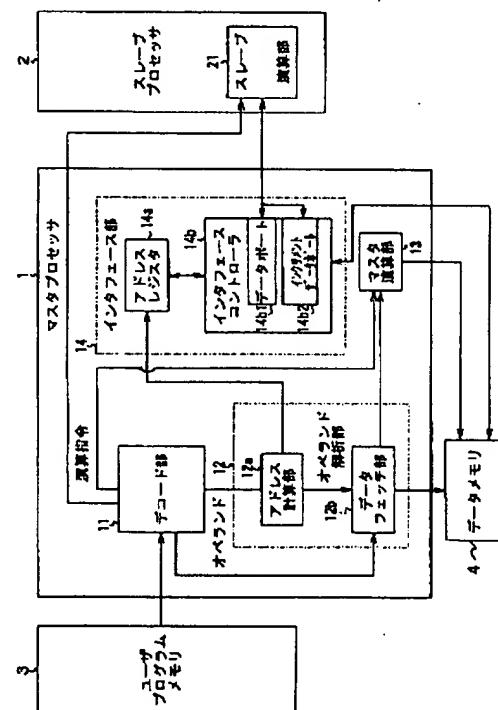
(74)代理人 弁理士 和田 成則

(54)【発明の名称】マルチプロセッサ演算装置、および該装置を有するプログラマブルコントローラ

(57)【要約】

【目的】マルチプロセッサ演算装置において、スレーブプロセッサ側における演算処理のオーバーヘッドを減少させる。

【構成】デコード部1 1は、各命令を解析して、その解釈に基づきマスタプロセッサ1側あるいはスレーブプロセッサ2側に演算指令を出力すると共に、当該命令のオペランドを出力する。アドレス計算部1 2 aは、そのオペランドに基づいてアドレスを計算して、アドレスレジスタ1 4 aに格納する。スレーブ演算部2 1がデコード部1 1より演算指令を受けて、インタフェース部1 4にアクセスすると、インタフェースコントローラ1 4 bは、アクセスレジスタ1 4 aに格納されたアドレスに基づき、スレーブ演算部2 1がデータメモリ4上におけるそのアドレスのデータへアクセスできるようにアクセス制御を行う。



【特許請求の範囲】

【請求項1】 各命令をマスタプロセッサ、あるいはスレーブプロセッサで実行するようにしたマルチプロセッサ演算装置であって、
上記マスタプロセッサは、

命令を解析すると共に当該命令がマスタプロセッサ側で実行されるものか、あるいはスレーブプロセッサ側で実行されるものかを判断して、その判断結果に基づきマスタプロセッサ側あるいはスレーブプロセッサ側に演算指令を出力すると共に、当該命令のオペランドを出力するデコード手段と、

上記デコード手段からのオペランドに基づいてアドレスを計算するアドレス計算手段と、

上記スレーブプロセッサ側からアクセス可能で、上記アドレス計算手段が計算したアドレスを記憶してそのアドレスのデータへのアクセスを制御するインターフェース手段と、を具備する一方、
上記スレーブプロセッサは、

上記デコード部からの演算指令を受けて上記インターフェース手段にアクセスして、当該インターフェース手段に記憶されたアドレスのデータを読み出し実行するスレーブ演算手段を具備する、
ことを特徴とするマルチプロセッサ演算装置。

【請求項2】 インタフェース手段は、

アドレス計算手段が計算したアドレスを記憶するアドレス記憶手段と、

上記アドレス記憶手段に記憶されたアドレスのデータへの直接アクセスを示すデータポートを有し、スレーブ演算手段がこのデータポートへアクセスした場合、当該スレーブ演算手段に上記アドレスが示すデータへ直接アクセスさせるよう制御するアクセス制御手段と、

を具備することを特徴とする請求項1記載のマルチプロセッサ演算装置。

【請求項3】 インタフェース手段は、

アドレス計算手段が計算したアドレスを記憶するアドレス記憶手段と、

上記アドレス記憶手段に記憶されたアドレスのデータへの直接アクセスを示すインクリメントデータポートを有し、スレーブ演算手段がこのインクリメントデータポートへアクセスした場合、当該スレーブ演算手段に上記アドレスが示すデータへ直接アクセスさせると共に、上記アドレス記憶手段に記憶されたアドレスをインクリメントするよう制御するアクセス制御手段と、

を具備することを特徴とする請求項1記載のマルチプロセッサ演算装置。

【請求項4】 請求項1、請求項2、または請求項3記載のマルチプロセッサ演算装置によってユーザプログラムを実行することを特徴とするプログラマブルコントローラ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、各命令をマスタプロセッサ、あるいはスレーブプロセッサで実行するようにしたマルチプロセッサ演算装置、および該装置を有するプログラマブルコントローラに関する。

【0002】

【従来の技術】プログラマブルコントローラ等のマルチプロセッサ演算装置では、通常、基本的な命令を高速実行するASIC等のマスタプロセッサと、それ以外の応用命令等を汎用MPU等によって実行するスレーブプロセッサ（1台でも、複数台でも勿論構わない。）とから構成されている。

【0003】マスタプロセッサは、命令メモリからの命令を解読して、その命令がこのマスタプロセッサで実行されるものか、あるいはスレーブプロセッサで実行されるものを判断し、自プロセッサが実行可能な命令についてはこのマスタプロセッサで実行すると共に、スレーブプロセッサが実行するものについては当該スレーブプロセッサに対して当該命令をデータとして送出するようしている。

【0004】一方、スレーブプロセッサは、通常は休止等として、マスタプロセッサからスレーブプロセッサ側で実行すべき命令を受けた場合には、当該命令の解析を行って実行するようしている。

【0005】つまり、例えば実行する命令のオペランドが3つある場合、スレーブプロセッサはマスタプロセッサ側から命令を受けると、図6に示すように命令の解釈（ステップ100）、オペランド1のアドレス計算（ステップ110）、オペランド1のデータフェッチ（ステップ120）、オペランド2のアドレス計算（ステップ130）、オペランド2のデータフェッチ（ステップ140）オペランド1によるデータとオペランド2によるデータとの演算（ステップ150）、オペランド3によるディスティネーション（転送先）アドレスの計算（ステップ160）、ディスティネーションへの書き込み（ステップ170）、という手順で処理を行っている。

【0006】

【発明が解決しようとする課題】しかし、このような従来のマルチプロセッサ演算装置では、スレーブプロセッサ側でも、上述の図7に示すように、命令の解釈を行い、続いて各オペランドにしたがったアドレス計算をして、命令の演算に使用するソースデータを取得し、さらにディスティネーションアドレスを算出して、その後実際の演算に入るようになっていたため、スレーブプロセッサに演算リクエストが送達されてから実際に演算を開始するまでの処理に時間がかかり、スレーブプロセッサにおけるオーバヘッドが大きくなってしまう、という問題があった。

【0007】そこで、本発明はこのような問題に着目してなされたもので、スレーブプロセッサ側における演算

処理のオーバーヘッドを減少させることのできるマルチプロセッサ演算装置、および該装置を有するプログラマブルコントローラを提供することを目的とする。

【0008】

【課題を解決するための手段】上記目的を達成するため、請求項1記載の発明では、各命令をマスタプロセッサ、あるいはスレーブプロセッサで実行するようにしたマルチプロセッサ演算装置であって、上記マスタプロセッサは、命令を解析すると共に当該命令がマスタプロセッサ側で実行されるものか、あるいはスレーブプロセッサ側で実行されるものかを判断して、その判断結果に基づきマスタプロセッサ側あるいはスレーブプロセッサ側に演算指令を出力すると共に、当該命令のオペランドを出力し、アドレス計算手段がそのオペランドに基づいてアドレスを計算して、インターフェース手段がそのアドレスを記憶する。

10

【0009】また、請求項2記載の発明では、請求項1記載のマスタプロセッサ演算装置において、インターフェース手段は、アドレス計算手段が計算したアドレスを記憶するアドレス記憶手段と、上記アドレス記憶手段に記憶されたアドレスのデータへの直接アクセスを示すデータポートを有し、スレーブ演算手段がこのデータポートへアクセスした場合、当該スレーブ演算手段に上記アドレスが示すデータへ直接アクセスさせるよう制御するアクセス制御手段と、を具備することを特徴とする。

20

【0010】また、請求項3記載の発明では、請求項1記載のマルチプロセッサ演算装置において、インターフェース手段は、アドレス計算手段が計算したアドレスを記憶するアドレス記憶手段と、上記アドレス記憶手段に記憶されたアドレスのデータへの直接アクセスを示すインクリメントデータポートを有し、スレーブ演算手段がこのインクリメントデータポートへアクセスした場合、当該スレーブ演算手段に上記アドレスが示すデータへ直接アクセスさせると共に、上記アドレス記憶手段に記憶されたアドレスをインクリメントするよう制御するアクセス制御手段と、を具備することを特徴とする。

30

【0011】また、請求項4記載の発明では、請求項1、請求項2、または請求項3記載のマルチプロセッサ演算装置によってユーザプログラムを実行するプログラマブルコントローラであることを特徴とする。

【0012】

【作用】請求項1～4記載の発明では、マスタプロセッサ側で、デコード手段が命令を解析すると共に当該命令

がマスタプロセッサ側で実行されるものか、あるいはスレーブプロセッサ側で実行されるものかを判断して、その判断結果に基づきマスタプロセッサ側あるいはスレーブプロセッサ側に演算指令を出力すると共に、当該命令のオペランドを出力し、アドレス計算手段がそのオペランドに基づいてアドレスを計算して、インターフェース手段がそのアドレスを記憶する。

【0013】そして、スレーブプロセッサ側のスレーブ演算手段が、デコード部からの演算指令を受けてインターフェース手段にアクセスすると、インターフェース手段の制御によってそのインターフェース手段に記憶されたアドレスのデータを読み出し、そのデータを実行する。

【0014】特に、請求項2記載の発明では、インターフェース手段は、アドレス計算手段が計算したアドレスを記憶しておく、スレーブ演算手段がデータポートへアクセスした場合には、スレーブ演算手段にそのアドレスが示すデータへ直接アクセスさせるよう制御する。

【0015】また、請求項3記載の発明では、インターフェース手段は、アドレス計算手段が計算したアドレスを記憶しておく、スレーブ演算手段がインクリメントデータポートへアクセスした場合には、当該スレーブ演算手段にそのアドレスが示すデータへ直接アクセスさせると共に、そのアドレスをインクリメントするよう制御する。

【0016】

【実施例】以下、本発明に係るマルチプロセッサ演算装置、および本装置を有するプログラマブルコントローラの実施例を図面に基づいて説明する。

【0017】図1に、本発明に係るマルチプロセッサ演算装置の構成を示す。

【0018】このマルチプロセッサ演算装置は、プログラマブルコントローラに設けられるもので、複数（本実施例では、便宜上2台とする。）のプロセッサ、すなはちマスタプロセッサ1、およびスレーブプロセッサ2と、ユーザプログラムを格納したユーザプログラムメモリ3と、データメモリ4とを基本的構成要素として備えている。

【0019】マスタプロセッサ1は、デコード部11と、オペランド解析部12と、マスタ演算部13と、インターフェース部14とを有している。

【0020】ここで、デコード部11は、ユーザプログラムメモリ3から命令を読み出して解析すると共に当該命令がマスタプロセッサ1側で実行されるものか、あるいはスレーブプロセッサ2側で実行されるものかを判断して、その判断結果に基づきマスタプロセッサ1側あるいはスレーブプロセッサ2側に演算指令を出力すると共に、当該命令のオペランドを出力するように構成されている。なお、デコード部11は、基本命令等の高速演算が要求される命令は、原則としてマスタプロセッサ1側に実行させ、応用命令や後からユーザが作成した命令等

40

50

はスレーブプロセッサ2側実行させるように判断するものとする。

【0021】オペランド解析部12は、デコード部11からのオペランドによりアドレスを計算して後述するデータフェッチ部12bおよびインタフェース部14のアドレスレジスタ14aに常時出力するアドレス計算部12aと、デコード部11での判断結果に基づいて命令がマスタプロセッサ1側で実行される場合には、アクセス計算部12aが計算したアドレスに基づきデータメモリ4にアクセスしてデータを読み出し、マスタ演算部13へ送出するデータフェッチ部12bとから構成されている。なお、このデータフェッチ部12bは、デコード部11での判断結果に基づいて命令がマスタプロセッサ1側で実行される場合のみデータメモリ4からデータを読み出してマスタ演算部13へ送出するようにしても、デコード部11での判断結果に基づいて常時命令がデータを読み出してマスタ演算部13へ送出するようにしても良い。

【0022】マスタ演算部13は、本実施例では、ASIC等の特定の命令を高速演算するもので、デコード部11からマスタプロセッサ1側への演算指令を受けて、その演算指令に基づきオペランド解析部12のデータフェッチ部12bがデータメモリ4から読み出したデータを入力して演算実行するように構成されている。

【0023】インタフェース部14は、アドレス計算部12aで計算された命令のアドレスをスレーブプロセッサ2側で実行される命令のアドレスとして命令毎に上書きで記憶するアドレスレジスタ14aと、インタフェースコントローラ14bとから構成されている。

【0024】インタフェースコントローラ14bは、スレーブプロセッサ2側からアクセス可能なデータポート14b1およびインクリメントデータポート14b2を有しており、スレーブプロセッサ2がデータポート14b1へアクセスした場合は、レジスタ14aに記憶されたアドレスをアドレスバスに出力等してスレーブプロセッサ2がデータメモリ4へ直接アクセスできるようメモリアクセスの制御を行うと共に、スレーブプロセッサ2がこのインクリメントデータポート14b2へアクセスした場合は、アクセスレジスタ14aに記憶されたアドレスに基づいてスレーブプロセッサ2がデータメモリ4へ直接アクセスできるようメモリアクセスの制御を行うと共に、その後アクセスレジスタ14aに格納されたアドレスを1インクリメントするように構成されている。

【0025】一方、スレーブプロセッサ2は、命令のデコード部も備えてなく、マスタプロセッサ1のデコード部11から命令の演算指令を受けて、その演算指令にしたがって次に説明する図2に示す処理手順でインタフェースコントローラ14bのデータポート14b1あるいはインクリメントデータポート14b2にアクセスして演算実行するスレーブ演算部21を備えている。

【0026】図2に、本実施例のスレーブ演算部21における命令演算手順を示す。

【0027】本実施例のスレーブ演算部21は、マスタプロセッサ1側のデコード部11から演算指令を受けると、スレーブ演算処理を開始して、まずはその演算指令を解釈する(ステップ200)。例えば、その演算指令がデータポート14b1へのアクセスか、あるいはインクリメントデータポート14b2へのアクセスを指示しているかや、さらにはその命令のオペランドの数等を解釈する。尚、ここでは、オペランドが3つ、すなわち演算すべき命令のソースデータのアドレスを示す2つのソースオペランドと、その演算結果の転送先を示す1つのディスティネーションオペランドの合計3つのオペランドがあるものとする。

【0028】そして、スレーブ演算部21は、その解釈に基づき、データポート14b1あるいはインクリメントデータポート14b2のどちらか一方のポートへアクセスして、アドレスレジスタ14aに格納されたオペランド1のアドレスが示すデータメモリ4上へ直接アクセスし、まずはオペランド1のアドレスのデータをフェッチする(ステップ210)。

【0029】次いで、同様にポートアクセスしてオペランド2のアドレスのデータをフェッチして(ステップ220)、続いて両フェッチしたデータを演算指令にしたがって演算し(ステップ230)、3度目のポートアクセスによってその演算結果をアドレスレジスタ14aに格納されたオペランド3のアドレスが示すディスティネーションへ転送して(ステップ240)、この命令の演算処理を終了する。

【0030】従って、この図2に示す処理をスレーブプロセッサ2側で実行する本実施例によれば、図6に示す処理をスレーブプロセッサ2側で実行する従来技術と比較して、ステップ110、130のオペランド1、2の計算であるソースデータアドレスの計算、およびステップ160のオペランド3の計算であるディスティネーションアドレスの計算処理が不要となり、スレーブプロセッサ2側のオーバーヘッドが減少して、スレーブプロセッサ2側の負担が軽減されると共に、スレーブプロセッサ2側の高速演算が可能になる。

【0031】次に、このように構成されたマスタプロセッサ演算装置の動作を、具体的な命令を一例にして説明する。

【0032】図3に、具体的な命令の一例を示す。

【0033】この命令は、スレーブプロセッサ2側で演算実行されるADD(加算)命令31を示しており、演算内容を示す命令コード“ADD”と、第1～第3の3つのオペランドとで構成されている。

【0034】このADD命令31の動作内容は、「(第1オペランドの)12番地のデータと、(第2オペランドの)IR0+4番地のデータとを加算して、その加算

結果を（第3オペランドの）18番地のデータが示す番地のデータに書き込む。」、という内容を示している。

【0035】なお、IR0、すなわちユーザプログラムメモリ3の0番地には、定数“10”的データが入っているものとする。

【0036】図4に、このADD命令31実行前のデータメモリ4のデータ格納状態の一例を示す。

【0037】このデータメモリ4には、例えば“8番地”的アドレスには定数“8”的データ、“11番地”的アドレスには定数“1”的データというように、アドレス順にデータが格納されている。 10

【0038】以上のメモリ状態を前提とした上で、この図2に示すADD命令31がマスタプロセッサ1のデコード部11によってユーザプログラムメモリ3から読み出されて解析されると、デコード部11からスレーブプロセッサ2側にそのADD命令31の演算内容を示す演算指令が送出されると共に、このADD命令31の第1～第3のオペランド（図2参照）がオペランド解析部12に送出される。

【0039】オペランド解析部12では、この第1～第3のオペランドをアドレス計算部12aが受けて、以下のようにアドレス計算を行う。 20

【0040】つまり、まず第1オペランドの場合、“\$12”的直接アドレス指定であるため、アドレス計算結果はそのままの定数“12”であり、これをアドレスレジスタ14aに書き込む。

【0041】また、第2オペランドの場合、“#4(IR0)”の間接アドレス指定であるため、IR0の内容“10”を読み込むと共に、“IR0+4”的演算を実行して“14”を得て、この計算結果“14”をアドレスレジスタ14aに書き込む。 30

【0042】また、第3オペランドの場合、“\$18”的直接アドレス指定であるため、アドレス計算結果はそのままの定数“18”であり、これをアドレスレジスタ14aに書き込む。

【0043】したがって、アドレスレジスタ14aには、このADD命令31の第1～第3のオペランドに各々対応した3つのアドレスが格納されることになる。

【0044】ところで、スレーブプロセッサ2では、マスタプロセッサ1側のデコード部11からそのADD命令31の演算指令をスレーブ演算部21が受けて、スレーブ演算部21がそのADD命令31の演算指令にしたがってインタフェースコントローラ14bのデータポート14b1若しくはインクリメントデータポート14b2にアクセスしてリードおよびライト動作を行う。 40

【0045】まず、スレーブ演算部21がデータポート14b1へアクセスしてリード動作をした場合、インタフェースコントローラ14bは、アドレスバスにアドレスレジスタ14に格納された各アドレスを出力等することによりスレーブ演算部21のメモリアクセスをコント 50

ロールして、アドレスレジスタ14aに格納された第1オペランドのアドレスである“12”をアドレスレジスタ14aから読出してアドレスバスに流す。

【0046】すると、図4に示すようにデータメモリ4のアドレス“12番地”的データ内容である定数“2”が、リードデータとしてデータメモリ4からデータポート14b1を介しスレーブプロセッサ2内に読み込まれることになる。

【0047】また、同様にして、スレーブ演算部21は、データポート14b2を再度アクセスするだけで、マスタプロセッサ1側で計算された第2オペランドのアドレス“14番地”的データである定数“4”をデータメモリ14から読込むことができる。

【0048】そして、スレーブ演算部21は、第1オペランドのアドレスに基づく定数“2”と、第2オペランドのアドレスに基づく定数“4”とを、デコード部11からのADDの演算指令にしたがって加算して定数“6”を得て、今度はその演算指令にしたがってデータポート14b1にアクセスして演算結果“6”的ライト動作を行う。

【0049】すると、データポート14b1にライトアクセスしただけで、インタフェースコントローラ14bのメモリアクセスコントロールによって、アドレスレジスタ14aに格納された第3オペランドのアドレス“8番地”がアドレスバスに出力され、データメモリ4のアドレス“8番地”へ演算結果の定数“6”が書き込まれることになる。

【0050】図5は、ADD命令31実行後のデータメモリ4の内容を示しており、図に示すように、データメモリ4のアドレス“8番地”的内容が定数“6”に変更されていることがわかる。

【0051】従って、本実施例によれば、図2にも示すように、スレーブプロセッサ2側では、命令のオペランドに基づいてアドレス計算をすることなく、マスタプロセッサ1側のインタフェースコントローラ14bのデータポート14b1にアクセスするだけで、インタフェースコントローラ14bのメモリアドレスコントロールによってデータメモリ4からデータを読み演算を実行して、ディスティネーションへ演算結果を書込むことができるようになる。

【0052】このため、図6に示すようにスレーブプロセッサ2側でも各オペランドに基づいてアクセス計算していた従来技術と比較して、ステップ110, 130のオペランド1, 2の計算であるソースデータアドレスの計算、およびステップ160のオペランド3の計算であるディスティネーションアドレスの計算処理が不要となり、スレーブプロセッサ2側のオーバーヘッドが減少し、スレーブプロセッサ2側の負担が軽減されると共に、スレーブプロセッサ2側の高速演算が可能になる。

【0053】次に、本実施例において、デコード部11

からの演算指令に基づいてスレーブ演算部21がインターフェースコントローラ14bのインクリメントデータポート14b2にアクセスした場合について説明する。

【0054】このインクリメントデータポート14b2へのアクセスは、デコード部11で解読された命令が、例えばデータメモリ4のある番地からある番地までの一定領域の各データを他の領域にブロック単位でコピーするデータのブロックコピー命令の場合や、一定領域の各データを各々演算して他の領域に書込む命令等のブロック単位の一括処理命令等の場合に行われる。

【0055】つまり、スレーブ演算部21がデコード部11からそのようなブロック単位でデータを処理する一括処理命令等の演算指令をうけたものとすると、スレーブ演算部21はインターフェースコントローラ14bのデータポート14ではなくインクリメントデータポート14b2にアクセスする。

【0056】このようなブロック転送命令の場合、アドレスレジスタ14aには、そのデータブロックの先頭番地が格納されているので（例えばアドレス“9番地”が格納されているものとする。）、スレーブ演算部21がインクリメントデータポート14b2にアクセスすると、インターフェースコントローラ14bがその先頭番地“9番地”的アドレスをアドレスバスに出力して、スレーブ演算部21にインクリメントデータポート14b2を介してデータメモリ4に直接アクセスさせ、その“9番地”的データである定数“9”を読み込み、ブロック単位のコピー等の演算処理を実行させることができる。

【0057】その際、その演算処理が終了すると、インターフェースコントローラ14bは、インクリメントデータポート14b2へのアクセスであるため、アドレスレジスタ14aに記憶された各オペランドのアドレスを1インクリメントする。

【0058】すると、スレーブ演算部21が再度インクリメントデータポート14b2にアクセスした場合、インターフェースコントローラ14bの制御によって、スレーブ演算部21がアドレスレジスタ14aに格納された1インクリメントされたアドレス“10番地”的データにアクセスでき、その再度のインクリメントデータポート14b2のアクセスによってアドレスレジスタ14aに格納されたアドレスが再度1インクリメントされ、次のアクセスの際には次のアドレス“11番地”的データにアクセスできることになる。

【0059】従って、本実施例によれば、スレーブ演算部21は、必要なデータ数分だけインクリメントデータポート14b2へのアクセスを繰り返すだけで、アドレス計算をすることなく一定領域のデータを読み出して実行することができ、一定領域のデータの一括処理の場合でもアドレス計算をせずに高速演算が可能になる。

【0060】

【発明の効果】以上説明したように本発明では、アドレ

ス計算手段が命令のオペランドに基づいて計算したアドレスをインターフェース手段が記憶しておき、スレーブプロセッサ側のスレーブ演算手段がインターフェース手段にアクセスした場合には、インターフェース手段はスレーブ演算手段がその記憶したアクセスのデータへアクセスできるようにアクセスコントロールを行うようにしたため、スレーブプロセッサ側では、命令のオペランドに基づいてアドレス計算をすることなく、データメモリからデータを読み込み実行することができる。

【0061】このため、スレーブプロセッサ側でも各オペランドに基づいてアクセス計算していた従来技術と比較して、ソースデータアドレスおよびディスティネーションアドレスの計算処理等が不要となり、スレーブプロセッサ側のオーバーヘッドが減少して、スレーブプロセッサ側の負担が軽減されると共に、スレーブプロセッサ側の高速演算が可能になる。

【0062】また、本発明では、スレーブ演算手段がインクリメントデータポートへアクセスした場合には、当該スレーブ演算手段にそのアドレスが示すデータへ直接アクセスさせると共に、そのアドレスをインクリメントするようにしたため、スレーブ演算手段が必要なデータ数分だけインクリメントデータポートへアクセスを繰り返すだけで、アドレス計算をすることなく一定領域のデータを読み出して実行することができ、一定領域のデータの一括処理の場合でもアドレス計算をせずに高速演算が可能になる。

【図面の簡単な説明】

【図1】本発明に係るマルチプロセッサ演算装置の第1実施例を示すブロック図。

【図2】第1実施例のスレーブ演算部における命令演算処理の手順を示すフローチャート。

【図3】具体的な命令を示す説明図。

【図4】命令実行前のデータメモリのデータ格納状態の一例を示す説明図。

【図5】命令実行後のデータメモリのデータ格納状態の一例を示す説明図。

【図6】従来のスレーブプロセッサにおける命令演算処理の手順を示すフローチャート。

【符号の説明】

1 マスタプロセッサ

1 1 デコード部（デコード手段）

1 2 オペランド解釈部

1 2 a アドレス計算部（アドレス計算手段）

1 2 b データフェッチ部

1 3 マスタ演算部

1 4 インタフェース部（インターフェース手段）

1 4 a アドレスレジスタ（アドレス記憶手段）

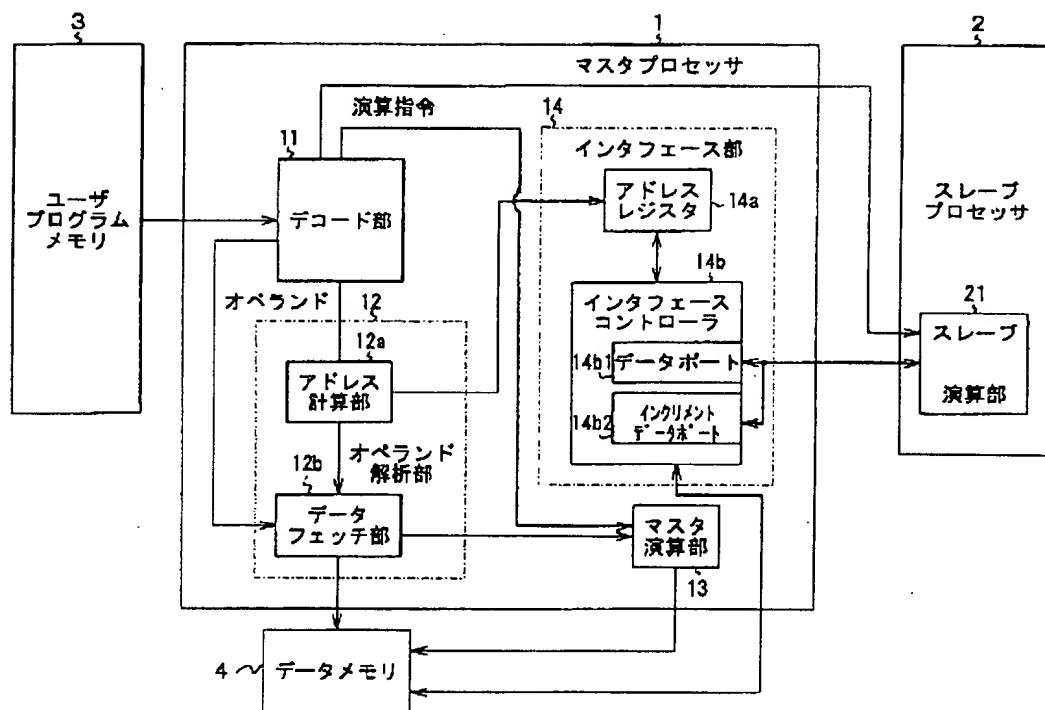
1 4 b インタフェースコントローラ（アクセス制御手段）

2 スレーブプロセッサ

2 1 スレーブ演算部
3 ユーザプログラムメモリ

4 データメモリ

【図1】



【図3】

31.ADD命令			
ADD (命令コード)	\$12, (第1オペランド)	#4 (R0), (第2オペランド)	(S18) (第3オペランド)

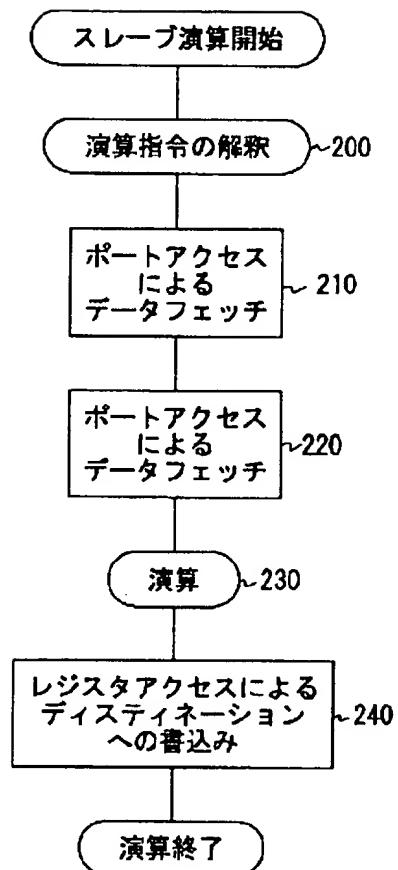
【図4】

4 データメモリ	
アドレス	データ
...	...
8番地	8
9番地	9
10番地	0
11番地	1
12番地	2
13番地	3
14番地	4
15番地	5
16番地	6
17番地	7
18番地	8
...	...

【図5】

4 データメモリ	
アドレス	データ
...	...
8番地	6
9番地	9
10番地	0
11番地	1
12番地	2
13番地	3
14番地	4
15番地	5
16番地	6
17番地	7
18番地	8
...	...

【図2】



【図6】

